JOURNAL OF INTERNATIONAL CRISIS AND RISK COMMUNICATION RESEARCH
ISSN: 2576-0017
2026, VOL 9, NO 5

# Mitigating Out-Of-Sequence Packet Drops In Stateful Ipsec Deployments Through Adaptive Sequence Number Management And Selective Rekeying

**Arun Raj Kaprakattu**

*Periyar University, India*

## Abstract
Counter drift remains an unresolved operational challenge in stateful IPsec redundancy. Each ESP or AH datagram carries a unique sequence value, but periodic state replication keeps the standby gateway perpetually a few values behind the active. Should the active fail in the gap between two replication ticks, the standby inherits an obsolete view of the counter. Traffic forwarded after promotion lands behind the remote peer's acceptance window and is silently rejected. Presented in this work is a self-contained corrective measure run by the freshly active gateway itself. Two coordinated actions form the core: the outbound counter is bumped forward by a magnitude derived from observed history, and renegotiation is initiated only on tunnels where this bump would push the counter past the value space the protocol permits. The mechanism avoids rekey storms, preserves anti-replay checks, and requires no awareness from the remote peer.

**Keywords:** IPsec high availability; ESP sequence continuity; anti-replay management; targeted SA rekeying; stateful failover recovery

### 1. Introduction
Encrypted IP communication over hostile networks relies on the IPsec protocol family [1], which combines payload confidentiality with origin verification. Within the deployment under study, a regular non-redundant gateway terminates tunnels whose remote endpoint is a redundant pair operating under stateful failover. Every transmitted ESP or AH packet stamps a sequence value into its header. Anti-replay logic [2] consults this value at the receiver to filter out duplicates and forged retransmissions. Strict in-order delivery is impractical on routed paths, so receivers tolerate reordering through a moving window of acceptable values [3]. Figure 1 depicts the geometry: anything below the window's trailing edge is rejected, while values past the leading edge slide the window forward. Continuity in the sequence stream breaks the moment the redundant side fails over. Packets emitted by the newly promoted gateway carry numbers older than what the standalone peer will accept, and these silent rejections continue until the gateways complete renegotiation (Child SA in IKEv2 [4][5], Phase 2 in IKEv1 [6]) or rekeying resets the counter into an accepted range.

### 1.1 Estimating Packet Loss Exposure During Failover
Quantifying maximum exposure begins with a configured replication period spanning active and standby units [7]. Worst case arises when failure occurs immediately preceding a scheduled state push, leaving the secondary unit operating from snapshot data taken almost a full cycle earlier. Multiplying per-tunnel throughput by tunnel count and elapsed interval yields the upper bound on packets potentially lost [1]. The resulting estimate grows alarmingly on aggregation hardware concentrating heavy traffic. Variability in actual losses depends on three live parameters: how long the replication window happens to be, what each tunnel was carrying,

and how many tunnels were active when the fault occurred. Notably absent from the IPsec specification is any feedback channel through which the standalone gateway could alert the redundant pair about ongoing rejection. The remainder of this work develops a one-sided remedy that allows the newly promoted gateway to handle the entire correction without involving the standalone counterpart in detection or signaling.

### 1.2 State Data Replicated Between Active and Standby Gateways

A wide range of state objects flow from active to standby during normal operation: IKE SAs, IPsec SAs, replay tracking counters, and IKE-layer message identifiers all need to remain mirrored [7]. Service continuity through a failover hinges on this mirroring. There is, however, a structural mismatch built into the design—replication is sampled at intervals while the live counter advances continuously. The unavoidable consequence is that the standby's ESP/AH sequence record consistently lags the active's, and that lag is the seed from which the drop scenario discussed here grows.

## 2. Deployment Scenarios Producing Out-of-Sequence Drops at the Standalone Peer Due to Stateful Switchover of the Redundant Peer

### 2.1 Dual-Chassis Stateful Redundancy Configuration

Periodic export of state from the primary chassis to its peer underpins inter-chassis redundancy [1][7]. Each export defines an interval; throughout that interval the peer's mirrored counter falls behind the live one. What makes the sequence number especially vulnerable is its update cadence—there is no batching, no tolerance, no grace period; every single packet ticks it forward. Network latency between the chassis pair widens the lag further. Sustained high packet rates cause divergence to balloon between exports, and the drop count following promotion of the secondary scales linearly with both the export interval and the offered load.

### 2.2 Intra-Chassis Line Card Level Redundancy Configuration

Internal redundancy at the line card tier mirrors the chassis-pair situation, with one architectural difference: replication traffic stays inside the box on the backplane [1]. Although that path delivers tighter, more uniform timing than any external network connection, it does not fully eliminate divergence—cards moving large packet volumes can still build a substantial counter gap inside a single replication tick. Underneath, the same dynamic is at play as in the multi-chassis case: a fixed export cadence guarantees an exposure window, and any failover landing inside that window produces out-of-window traffic. Speed and predictability of the backplane shift the numbers but do not change the underlying mechanism.

**Table 1: Comparison of Synchronization Scenarios in Stateful IPsec Redundancy [1, 7]**

| HA Configuration | State Propagation Direction | Replication Transport | Replication Latency Profile | Consistency Risk at Failover |
|---|---|---|---|---|
| Dual-Chassis HA | Primary chassis replicates to secondary chassis | Dedicated inter-chassis link or routed path | Elevated and variable; governed by inter-chassis link conditions | Counter divergence grows with link latency and replication period; worst-case loss proportional to both |

| | | | | |
|---|---|---|---|---|
| Intra-Chassis Line Card HA | Primary line card replicates to redundant line card | Internal backplane fabric | Reduced and predictable; backplane path is bounded | Divergence scales with per-card packet rate and replication cadence |

## 3. Analysis of Current Approaches to Post-Switchover Sequence Drop Recovery

### 3.1 Global Phase 2 Renegotiation Triggered by the Newly Active Device

A widely deployed countermeasure schedules a fresh Phase 2 negotiation across all surviving tunnels as soon as the new active device starts forwarding, with no distinction made between line card and chassis failovers [8]. New Security Associations bring counters back into agreement on both sides. What undoes this approach in practice is volume. Carrier-grade aggregators can be holding open millions of concurrent tunnels [1], and pushing the entire population through key exchange simultaneously generates a control-plane spike that the gateway often cannot absorb. Symptoms cascade: IKE queues lengthen, individual rekey transactions slow, forwarding capacity suffers, and a fraction of tunnels can fail to come back at all. A mechanism designed to make failover invisible can paradoxically extend and deepen the outage it was meant to mask.

### 3.2 Relaxing or Removing the Anti-Replay Window Constraint

Bypassing the anti-replay test outright, or stretching the acceptance bitmap to span virtually all possible counter values, halts the rejections at once [9]. When verification is switched off, every inbound datagram is admitted without regard to its sequence stamp. Enlarging the bitmap produces the same effect by inflating the range deemed in-window. Both routes purchase availability with a security debit. Wider bitmaps mean a replayed datagram can remain undetected in transit longer before discovery, eroding the very property anti-replay was added to enforce [3]. Switching the check off entirely strips the tunnel of replay protection altogether. There is a second cost less often noted: when out-of-order delivery is admitted without bound, jitter rises—and applications such as voice telephony or live video, which assume orderly arrival, suffer perceptible degradation.

### 3.3 Targeted Rekeying Driven by the Non-Redundant Remote Peer

Yet another design relocates the burden onto the non-redundant gateway. By monitoring its own arrival stream for evidence that the remote counter has discontinuously regressed, it attempts to recognize a probable switchover and then issues rekey messages only on the tunnels showing symptoms [10]. Limiting renegotiation scope this way is appealing in principle. Implementation is the difficulty. From its position the standalone gateway possesses no privileged knowledge about the redundancy mode of its peer. Inference must proceed entirely from observable traffic, where ordinary reordering and a true counter rollback can look indistinguishable for non-trivial periods. Producing detection logic that is both fast and free of false alarms under realistic conditions has resisted previous attempts.

**Table 2: Analysis of Existing Solutions for Out-of-Sequence Traffic Recovery [8, 9, 10]**

| Recovery Method | Triggering Entity | Affected Tunnels | Control-Plane Cost | Security Effect | Traffic Impact |
|---|---|---|---|---|---|
| Global Phase 2/Child SA Rekey [8] | Newly promoted active device | Every established tunnel | Severe burst; all tunnels negotiate concurrently | Security posture preserved throughout | Data-plane disruption proportiona |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | l to tunnel count |
| Disable/Expand Anti-Replay Window [9] | Configuration change on local peer | All flows on involved tunnels | Near zero; no key exchange required | Replay defense removed; injection risk present | Out-of-order delivery degrades latency-sensitive traffic |
| Remote-Initiated Selective Rekey [10] | Non-redundant remote device | Impacted tunnels only | Moderate; detection logic adds overhead | Anti-replay posture unchanged | Drops persist until detection threshold is crossed |

## 4. The Proposed Solution

The proposed mechanism comprises two interlocking components. As soon as the secondary is promoted, it derives a numerical offset from its most recent replicated snapshot and applies that offset to the outbound counter before the first datagram leaves the gateway [2][11]. With this preadjustment in place, the very first packet arriving at the remote peer falls inside its acceptance window and is admitted normally. No knowledge or action is needed at the remote end. The second piece is a per-tunnel guard: each candidate tunnel is checked to see whether the offset would push its counter past the protocol-defined ceiling. Only those that would are queued for Phase 2 or Child SA negotiation; the remainder resume forwarding without interruption. By dividing labor this way the design avoids both pitfalls of prior approaches—mass renegotiation and weakened replay protection.

### 4.1 Proactive Counter Advancement on the New Active Device

A receive-side window is held per Security Association. Its top boundary tracks the largest legitimately received sequence value seen on that SA so far. Whenever a packet arrives carrying a value that surpasses this boundary, the entire window slides forward to incorporate the new top. After such a slide, anything older than the recomputed bottom boundary is classed as either replayed or simply invalid and discarded immediately.

Receipt of a counter higher than the present upper bound is not treated as an error. Instead the window glides ahead until the freshly arrived value sits at its leading edge. As long as the value remains within protocol-permitted range and has not wrapped past the 32-bit or 64-bit ceiling, the datagram is admitted. Any older entries already tracked in the window structure that fall outside the new trailing edge are simply expired from the replay tracking bitmap.

Counter advancement has a single overarching purpose: cover the entire range of sequence numbers the failed active device may have used between its last replication push and its failure. With that interval bridged, the first packet from the replacement gateway lands inside the remote peer's active window rather than trailing behind it.

Two inputs drive the offset calculation: typical throughput, and the amount of variance present in arrival timing. A baseline increment is read from the largest counter delta seen across the most recent replication intervals; this constitutes a safe overestimate of how far the counter could realistically have advanced since the last snapshot. Multiplying this baseline by a jitter compensation coefficient adds margin in case actual burst behavior exceeds anything previously observed. The coefficient itself is exposed for operator configuration, letting deployments calibrate the mechanism to whatever traffic profile they actually run—tight enterprise concentrators at one end, and bursty service-provider edges at the other.

Edge case handling: if M is zero or uninitialized—as may occur on newly provisioned tunnels or very low-traffic SAs—a configurable minimum floor value M_min should be substituted to

prevent a zero-advance scenario. For the burst scaling multiplier X, we recommend a practical starting range of 1.5 to 3.0: stable enterprise concentrators can operate at the lower end, while bursty service-provider edges should use larger values. The explicit formula for the post-switchover counter seed is: $S_n = S_0 + (M \times X)$, where M = max(peak inter-sync increment, M_min).

At the moment forwarding resumes on the promoted gateway, its outbound counter is initialized to the last replicated value plus the calculated offset. From there, each successive packet increments the counter normally. Because the seeded value is already comfortably inside the remote peer's acceptance range, datagrams are admitted on first arrival, and no configuration tweak or protocol extension is required at the standalone side to make this work.

Note on Extended Sequence Numbers (ESN): The counter advancement mechanism described above applies equally to both 32-bit and 64-bit (ESN) sequence number spaces as defined in RFC 4304 [11]. With ESN, the protocol ceiling is $2^{64} - 1$, making overflow-triggered rekeying extremely rare in practice; nevertheless, the per-tunnel ceiling check in Section 4.2 is applied identically regardless of sequence number width, ensuring correctness in both modes.

**Table 3: Sequence Number Management Parameters in Proposed Solution [1, 2, 7]**

| Parameter | Definition | How It Is Obtained | Configuration Mode | Role in Recovery Process |
|---|---|---|---|---|
| Replicated Counter Baseline ($S_0$) | Counter value held in the most recent replicated snapshot | Extracted from the standby state database at switchover [7] | Automatically populated by replication engine | Starting point from which the delta is added |
| Peak Inter-Sync Increment (M) | Largest counter jump recorded across recent replication cycles | Derived by inspecting inter-replication interval counters [1] | Computed from measured traffic history | Sets the floor for the delta computation |
| Burst Scaling Multiplier (X) | Scaling factor applied to M to absorb burst traffic | Set according to deployment traffic profile | Exposed as an operator-adjustable parameter | Provides headroom against counter under-advancement |
| Counter Advancement Delta (D) | Total counter advancement applied at switchover | Product of maximum increment and compensation factor | Calculated from observed maximum and operator multiplier | Determines where the outbound counter is seeded |
| Post-Switchover Seed Value ($S_n$) | Initial counter value used by the new active device | Sum of last replicated value and computed delta | Automatically computed at switchover time | Guarantees first post-switchover packet is accepted by remote peer |

### 4.2 Targeted Phase 2 and Child SA Negotiation Triggering

A per-tunnel evaluation runs on the active gateway: the projected counter value after offset application is compared against the protocol-mandated upper ceiling [2][11]. Tunnels whose projected value sits safely below the ceiling are released to resume forwarding immediately. Tunnels whose projected value would cross or approach the ceiling are routed into

negotiation—Phase 2 under IKEv1, Child SA under IKEv2—through which the standard key exchange resets their counters cleanly. This narrow targeting keeps rekey volume tied directly to the tunnel population genuinely facing exhaustion, rather than to the gateway's total tunnel footprint. Forwarding throughput and IKE processing headroom are preserved for everything else.

## 5. Evaluation of the Proposed Approach Against Alternative Recovery Strategies

### 5.1 Mass Renegotiation Initiated by the New Active Device After Switchover to Avoid Out-of-Sequence Traffic

Blanket renegotiation pushes every tunnel through a fresh key exchange without regard to whether intervention is actually required [8][1]. Quiescent tunnels, low-rate tunnels, and tunnels whose state happened to have been replicated recently all end up consuming IKE cycles needlessly. Every avoidable transaction adds to the burst that risks toppling the gateway. By contrast, the proposed approach computes ahead of time which tunnels truly need attention, then directs renegotiation only at that subset. Control-plane cost tracks the size of the actual problem rather than the size of the total tunnel population. Unaffected tunnels—the vast majority in any realistic deployment—continue forwarding immediately and uninterrupted.

### 5.2 Anti-Replay Suppression as a Recovery Strategy

Two distinct penalties accrue from anti-replay suppression. On the data plane, allowing arbitrarily out-of-order delivery elevates jitter, which is felt by latency-sensitive workloads such as VoIP and video conferencing [12]. On the security plane, it strips the tunnel of the replay protection that virtually every modern compliance regime expects IPsec to deliver. Neither cost arises with the proposed mechanism. By ensuring outbound counters fall inside the remote window from the first packet onward, the underlying mismatch is corrected at its source [9], and anti-replay verification stays continuously enforced. Policy compliance is preserved without compromise, satisfying obligations that suppression would directly violate.

It is worth noting a transient condition introduced by counter advancement itself: when the promoted gateway seeds its outbound counter at $S_n = S_0 + D$, the remote peer's replay bitmap contains no entries for sequence values in the range $(S_0, S_n)$. An adversary who captured packets — with sequence numbers in that range — from the failed active device could inject them into the now-valid window immediately after failover. This exposure window is brief—bounded by the time until the remote bitmap fills the gap through normal forwarding—but operators in high-security environments should treat it as a known, transient risk.

### 5.3 Remote-Side Triggered Selective Renegotiation

Putting selective rekey responsibility on the standalone gateway demands that it identify the remote switchover event and act on it [10]. The proposed alternative dispenses with this requirement completely. Detection, offset application, and conditional renegotiation all reside on the redundant pair, and the standalone end takes no role in any of them. Recovery commences the instant the secondary assumes the active position, before the standalone gateway could have observed enough drops to even begin inferring a problem—so the detection-window losses that haunt reactive designs simply never accumulate. The standalone gateway needs no augmentation, recovery completes faster, and the architecture as a whole is leaner because the remediation work runs where the relevant state lives.

**Table 4: Comparative Analysis of Selective Versus Global Rekeying Approaches [2, 8, 9, 10 11]**

| Assessment Dimension | Proposed Selective Rekeying [2][11] | Global Rekeying [8] | Disabled Anti-Replay [9] | Remote-Initiated Selective [10] |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Rekeying Scope | Exclusively tunnels whose advanced counter nears the ceiling | Every established tunnel regardless of need | Rekeying entirely avoided | Tunnels flagged by drop detection logic |
| Control-Plane Burden | Low; scales with impacted tunnel count only | Peak; entire tunnel population negotiates at once | Near zero | Moderate; includes detection processing cost |
| Time to Full Recovery | Instant recovery for the unaffected majority | Recovery deferred until mass negotiation finishes | Fast but with reduced security guarantees | Recovery deferred pending drop detection |
| Drops During Recovery Window | Minimal; only affected tunnels experience interruption | Loss possible while mass negotiation is in progress | Ongoing; replayed packets accepted without limit | Drops accumulate until detection logic fires |
| Anti-Replay Status | Anti-replay enforcement uninterrupted [9] | Complete anti-replay coverage preserved | Weakened; replay attacks possible | Complete anti-replay coverage preserved |
| Switchover Detection Requirement | Unnecessary; correction applied before traffic flows | None needed | N/A | High; relies on reliable switchover inference |
| Scaling Behavior | Scales well; cost decoupled from total tunnel count | Does not scale; cost grows with full tunnel population | Scalable but security-compromised | Reasonable; cost tied to impacted tunnel count |
| Deployment Complexity | Moderate; requires per-tunnel counter evaluation | Minimal; no per-tunnel analysis needed | Minimal; single configuration flag | High; detection and coordination logic required |

**Conclusion**

Among the failure modes intrinsic to stateful IPsec redundancy, out-of-window drops following a switchover have not been adequately addressed by existing remedies. A two-stage mechanism described in this paper resolves the problem upstream of where it manifests. Counter preadjustment ensures the remote peer never sees a single out-of-window datagram in the first place. Conditional renegotiation handles only those tunnels where preadjustment alone would breach the protocol's counter ceiling. Operating together, these stages restore continuity without inflicting either the control-plane spike of universal rekeying or the security regression of switching anti-replay off.

A known limitation of the present design concerns rapid successive failovers. If the newly promoted gateway itself fails before completing a full replication cycle, its successor will inherit the pre-seeded counter value $S_n$ rather than the true live counter. In this scenario the offset calculation at the third device may over-advance the counter. Operators should be aware of this edge case in environments where back-to-back chassis failures are plausible; multi-fault behavior is identified as a direction for future work.

Historical replication intervals together with observed packet rates feed the offset computation, producing a delta sized to span the gap between the snapshot the standby holds and the value the active was likely on at failure. That offset is applied by the promoted gateway prior to its first transmission. Service is restored at the moment of promotion, with no dependence on the remote peer perceiving drops or on any threshold being crossed in detection logic. As a result, packet loss is confined to the switchover instant itself and does not stretch across an extended detection-response interval.

Negotiation activity in the proposed scheme is bounded by need: only the tunnels whose offset-shifted counters would breach the protocol ceiling enter renegotiation, ensuring rekey volume tracks actual demand. Earlier mitigations forced operators to pick between drowning the control plane or weakening security; the present approach refuses that dichotomy by correcting the underlying counter gap proactively, leaving unnecessary rekeys to be the exception rather than the routine response. The savings become significant at large scale, where deployments concentrate millions of concurrent tunnels on a single piece of infrastructure.

Across every dimension assessed in Table 4, the proposed mechanism comes out ahead. Anti-replay enforcement runs continuously, so the security expectations that other approaches relax remain fully satisfied. Compute cost on the gateway tracks the cardinality of affected tunnels, not the cardinality of the entire tunnel population. Decoupling overhead from gateway scale is the signature property of this design, and the advantage grows wider as concentration density increases—making the technique a strong fit for the largest enterprise and carrier deployments.

Modifications to the standalone gateway are unnecessary. Every part of the corrective workflow runs on the redundant pair, which is also the only side in possession of the operational state required to drive it. Inter-peer interaction therefore remains entirely within standard IPsec and IKE semantics, and the deployment story stays simple: only one end of every tunnel needs new code or configuration.

Tuning of the mechanism centers on a single knob: the jitter compensation coefficient. Workloads that are stable and predictable, such as those terminating on enterprise concentrators, can run comfortably with a small multiplier. Mixed and bursty workloads typical of service provider edges call for a larger value to accommodate spikes. The algorithmic core itself stays unchanged across environments—only this coefficient is varied to fit the deployment.

There are multiple directions that future work could profitably explore. Substituting the conservative max-increment baseline with a predictively modeled estimate could yield a tighter delta and reduce counter overshoot. Dynamic compensation, in which the multiplier adjusts to live measurements, could replace the static configuration model used here. Validation under multi-fault scenarios—rapid back-to-back switchovers, concurrent line card failures, and cascading failover chains—would confirm the design's behavior outside the single-switchover assumption that frames the present analysis. Tracking compatibility with newer IPsec protocol extensions as the standards continue to develop is also worth ongoing attention.

Pairing preadjustment of the outbound counter with selective, criterion-driven renegotiation produces a recovery design that goes easy on control-plane resources, scales to large tunnel populations without performance penalty, and keeps the anti-replay guarantee built into IPsec entirely intact. Together these characteristics make the design suitable for production high-availability environments—the kind in which neither security policy can yield ground nor service continuity can be put at risk.

**References**

1. Akiko Kuboniwa, et al., "Ipsec-gw redundancy method with high reliability," in 8th Asia-Pacific Symposium on Information and Telecommunication Technologies, pages 1–5, IEEE, 2010.https://ieeexplore.ieee.org/document/5532017
2. Stephen Kent, "Rfc 4303: Ip encapsulating security payload (esp)," Technical report, 2005.https://datatracker.ietf.org/doc/html/rfc4303
3. Fan Zhao, et al., "Analysis and improvement on ipsec anti-replay window protocol," in Proceedings. 12th International Conference on Computer Communications and Networks

(IEEE Cat. No. 03EX712), pages 553–558, IEEE, 2003. https://ieeexplore.ieee.org/document/1284223
4. Charlie Kaufman et al., "RFC 7296: Internet Key Exchange Protocol Version 2 (IKEv2)," Technical report, 2014.https://www.rfc-editor.org/rfc/rfc7296.htmlhttps://www.rfc-editor.org/rfc/rfc4306.html
5. CJ Tjhai, et al., "Rfc 9370: Multiple key exchanges in the internet key exchange protocol version 2 (ikev2)," Technical report, 2023.https://datatracker.ietf.org/doc/html/rfc9370
6. Dan Harkins, et al., "Rfc2409: The internet key exchange (ike)," Technical report, 1998. https://datatracker.ietf.org/doc/html/rfc2409
7. R Singh, et al., "Protocol support for high availability of ikev2/ipsec," Technical report, 2011. https://datatracker.ietf.org/doc/html/rfc6311
8. Daniel Palomares, et al., "High availability for ipsec vpn platforms: Clusterip evaluation," in 2013 International Conference on Availability, Reliability and Security, pages 178–187, IEEE, 2013. https://ieeexplore.ieee.org/document/6657239
9. Mohamed G Gouda Chin-Tser Huang, et al., "Anti-replay window protocols for secure ip," Technical report. 06 August 2002. https://ieeexplore.ieee.org/document/885507
10. Arun Raj Kaprakattu, "Ipsec tunnel recovery from out-of-sequence traffic drop due to peer ipsec stateful switchover," June 2025. Journal of Computer and Communications, 13(6):26–32, 2025. https://www.scirp.org/journal/paperinformation?paperid=143251
11. Stephen Kent, "Extended sequence number (esn) addendum to ipsec domain of interpretation (doi) for internet security association and key management protocol (isakmp)," Technical report, 2005. https://www.rfc-editor.org/rfc/rfc4304.html
12. K. Chan et al., "Configuration Guidelines for DiffServ Service Classes," Technical report, 2006. https://datatracker.ietf.org/doc/html/rfc4594https://datatracker.ietf.org/doc/html/rfc3948